2.STUDENT DETAILS WITH SWAGGERUI:

CONTROLLER:

package com.example.color.studentcontroller2;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.CrossOrigin;

import org.springframework.web.bind.annotation.DeleteMapping;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.PutMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RestController;

import com.example.color.studententity2.studentent;

import com.example.color.studentservice2.studentser;

@CrossOrigin

@RestController

public class studentcont {

@Autowired

studentser stuservice;

@PostMapping("post")

public studentent addInfo(@RequestBody studentent st) {

return stuservice.saveDetails(st);

}

@GetMapping("get")

public List<studentent> fetchDetails(){

return stuservice.getDetails();

}

@PutMapping("update")

public studentent updateInfo(@RequestBody studentent st1) {

return stuservice.updateDetails(st1);

}

@DeleteMapping("/delete/{id}")

public String deleteInfo(@PathVariable("id") int id) {

stuservice.deleteDetails(id);

return "Deleted Details";

}

}

ENTITY:

package com.example.color.studententity2;

import jakarta.persistence.\*;

@Entity

@Table(name="student")

public class studentent {

@Id

private int studentid;

private String studentname;

private int age;

public int getStudentid() {

return studentid;

}

public void setStudentid(int studentid) {

this.studentid = studentid;

}

public String getStudentname() {

return studentname;

}

public void setStudentname(String studentname) {

this.studentname = studentname;

}

public int getAge() {

return age;

}

public void setAge(int age) {

this.age = age;

}

public studentent(int studentid, String studentname, int age) {

super();

this.studentid = studentid;

this.studentname = studentname;

this.age = age;

}

public studentent() {

}

}

REPOSITORY:

package com.example.color.studentrepos2;

import org.springframework.data.jpa.repository.JpaRepository;

import com.example.color.studententity2.studentent;

public interface studentrep extends JpaRepository<studentent,Integer>{

}

SERVICE:

package com.example.color.studentservice2;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import com.example.color.studententity2.studentent;

import com.example.color.studentrepos2.studentrep;

@Service

public class studentser {

@Autowired

studentrep stRepo;

public studentent saveDetails(studentent e) {

return stRepo.save(e);

}

public List<studentent> getDetails(){

return stRepo.findAll();

}

public studentent updateDetails(studentent e1) {

return stRepo.saveAndFlush(e1);

}

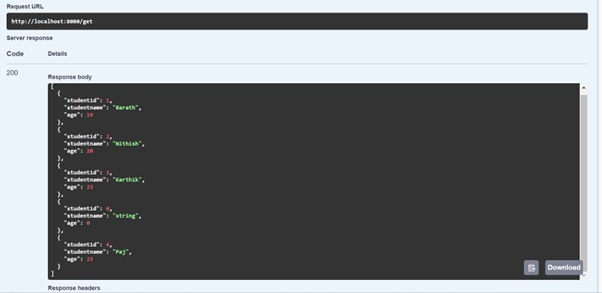
public void deleteDetails(int id) {

stRepo.deleteById(id);

}

}

OUTPUT:



3.GET BIKE DETAILS WITH JPQL QUERY:

CONTROLLER:

package com.example.demo.Controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.DeleteMapping;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.PutMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RestController;

import com.example.demo.Entity.StudentDetails;

import com.example.demo.Service.StudentService;

@RestController

public class StudentController {

@Autowired

StudentService stuService;

@PostMapping("post")

public StudentDetails addInfo(@RequestBody StudentDetails st) {

return stuService.saveDetails(st);

}

@GetMapping("get")

public List<StudentDetails> fetchDetails(){

return stuService.getDetails();

}

@PutMapping("/update")

public StudentDetails updateInfo(@RequestBody StudentDetails st) {

return stuService.saveDetails(st);

}

// @DeleteMapping("/delete")

// public void deleteInfo(@RequestBody StudentDetails st) {

// stuService.deleteDetails(st);

// }

@DeleteMapping("/delete/{sid}")

String deleteinfo() {

return "Deleted the element";

}

@GetMapping("/showAllInfo")

public List<StudentDetails> displayAllStudent(){

return stuService.allStudent();

}

@GetMapping("/showId/{sid}")

public List<StudentDetails> displayAllStudentbyId(@PathVariable("sid") int sid){

return stuService.allStudentbyId(sid);

}

@GetMapping("/showIdName/{sid}/{sname}")

public List<StudentDetails> displayAllStudent(@PathVariable("sid") int sid,@PathVariable("sname") String sname){

return stuService.allStudentbyIdName(sid,sname);

}

@DeleteMapping("/removeid/{sid}")

public Integer displayAllStudent(@PathVariable("sid") int sid){

return stuService.removeStudentbyId(sid);

}

@PutMapping("/putDetail/{sname}/{sid}")

public String update(@PathVariable("sname") String sname,@PathVariable("sid") int sid) {

return stuService.updateStudent(sname,sid)+" updated";

}

}

ENTITY:

package com.example.demo.Entity;

import jakarta.persistence.Column;

import jakarta.persistence.Entity;

import jakarta.persistence.Id;

import jakarta.persistence.Table;

@Entity

@Table(name="BikeDetails")

public class StudentDetails {

@Id

@Column(name="bikerid")

private int bikerid;

private String regnumber;

private String ownername;

private int year;

private String bikename;

private String modelname;

public int getBikerid() {

return bikerid;

}

public void setBikerid(int bikerid) {

this.bikerid = bikerid;

}

public String getRegnumber() {

return regnumber;

}

public void setRegnumber(String regnumber) {

this.regnumber = regnumber;

}

public String getOwnername() {

return ownername;

}

public void setOwnername(String ownername) {

this.ownername = ownername;

}

public int getYear() {

return year;

}

public void setYear(int year) {

this.year = year;

}

public String getBikename() {

return bikename;

}

public void setBikename(String bikename) {

this.bikename = bikename;

}

public String getModelname() {

return modelname;

}

public void setModelname(String modelname) {

this.modelname = modelname;

}

public StudentDetails(int bikerid, String regnumber, String ownername, int year, String bikename,

String modelname) {

super();

this.bikerid = bikerid;

this.regnumber = regnumber;

this.ownername = ownername;

this.year = year;

this.bikename = bikename;

this.modelname = modelname;

}

public StudentDetails() {

}

}

REPOSITORY:

package com.example.demo.Repository;

import java.util.List;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.data.jpa.repository.Modifying;

import org.springframework.data.jpa.repository.Query;

import org.springframework.data.repository.query.Param;

import com.example.demo.Entity.StudentDetails;

import jakarta.transaction.Transactional;

public interface StudentRepo extends JpaRepository<StudentDetails,Integer> {

//get all details from table

@Query(value="select \* from student\_details ",nativeQuery=true)

public List<StudentDetails> getStudentInfo();

//get specified detail by id

@Query(value="select \* from student\_details where sid=:s",nativeQuery=true)

public List<StudentDetails> getStudentbyId(@Param("s") int sid);

//get detail by name or id

@Query(value="select \* from student\_details where sid=:s or sname=:sn",nativeQuery=true)

public List<StudentDetails> getStudentbyIdName(@Param("s") int s,@Param("sn") String sn);

//delete detail

@Modifying

@Transactional

@Query(value="delete from student\_details where sid=:i",nativeQuery=true)

Integer deleteByStudentId(@Param("i") int sid);

//update name using id

@Modifying

@Transactional

@Query(value="update student\_details set sname=:sn where sid=:i",nativeQuery=true)

public Integer updatebynameid(@Param("sn") String sname,@Param("i") int sid);

}

SERVICE:

package com.example.demo.Service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import org.springframework.web.bind.annotation.PathVariable;

import com.example.demo.Entity.StudentDetails;

import com.example.demo.Repository.StudentRepo;

@Service

public class StudentService {

@Autowired

StudentRepo stRepo;

//CRUD Operations

public StudentDetails saveDetails(StudentDetails e) {

return stRepo.save(e);

}

public List<StudentDetails> getDetails(){

return stRepo.findAll();

}

public StudentDetails updateDetails(StudentDetails e1) {

return stRepo.saveAndFlush(e1);

}

public void deleteDetails(int id) {

stRepo.deleteById(id);

}

public List<StudentDetails> allStudent(){

return stRepo.getStudentInfo();

}

public List<StudentDetails> allStudentbyId(int a){

return stRepo.getStudentbyId(a);

}

public List<StudentDetails> allStudentbyIdName(int a,String b){

return stRepo.getStudentbyIdName(a,b);

}

public Integer removeStudentbyId(int a){

return stRepo.deleteByStudentId(a);

}

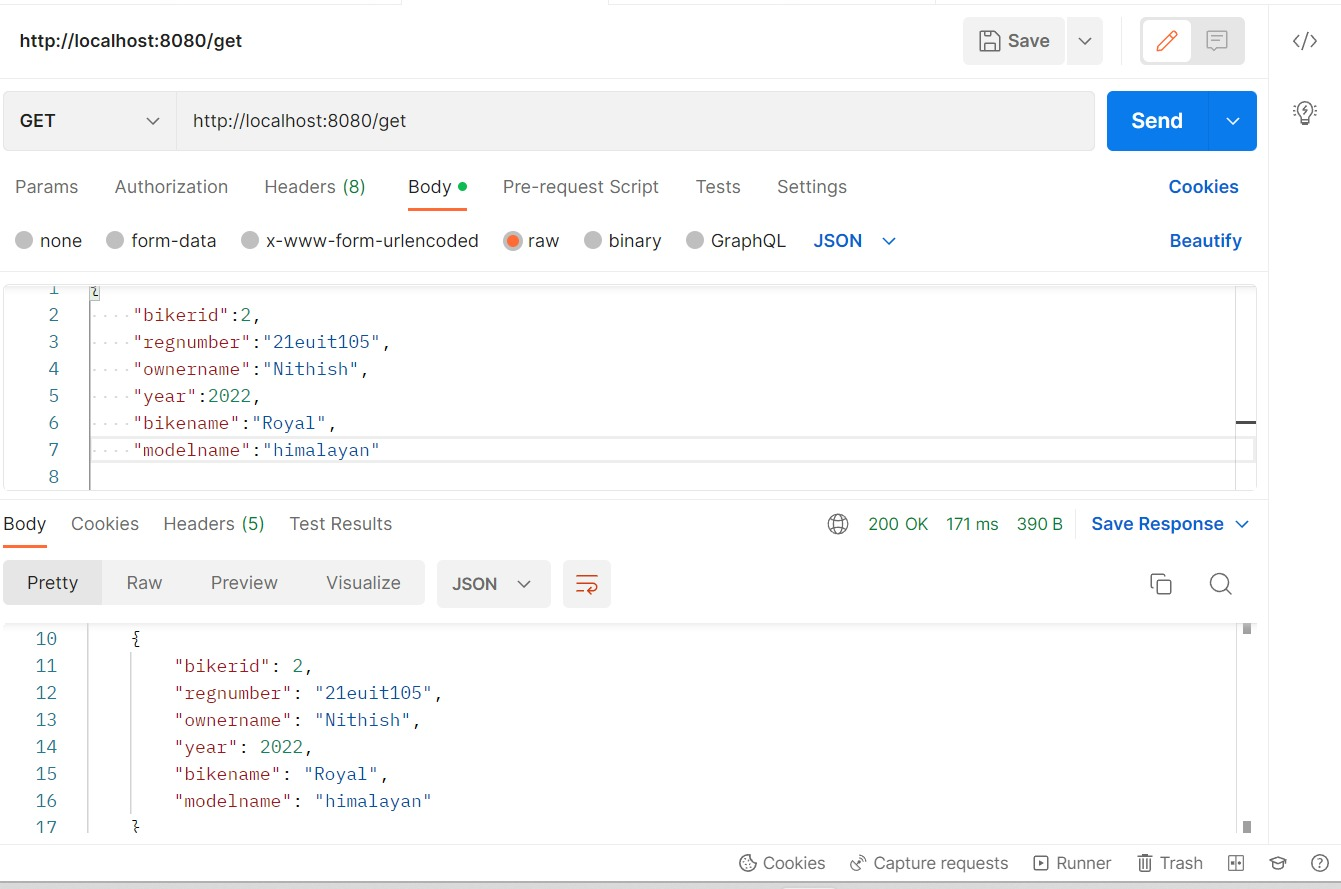
public Integer updateStudent(String sname,int sid) {

return stRepo.updatebynameid(sname, sid);

}

}

OUTPUT:



4.GET EMPLOYEE DETAILS WITH SORTING AND PAGINATION:

CONTROLLER:

package com.example.color.testcontroller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.DeleteMapping;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.PutMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RestController;

import com.example.color.testentity.testentity;

import com.example.color.testservices.testservices;

@RestController

public class testcontroller {

@Autowired

testservices s;

@GetMapping("/get")

public List<testentity> showDetails() {

return s.getDetails();

}

@PostMapping("/post")

public String addDetails(@RequestBody testentity m) {

s.add(m);

return "Added product " + m.getEmployeeid();

}

@PutMapping("/update")

public testentity updateInfo(@RequestBody testentity m2 ) {

return s.updateDetails(m2);

}

@DeleteMapping("/delete")

public void deleteInfo(@RequestBody testentity st2) {

s.deleteDetails(st2);

}

//sorting

@GetMapping("/product/{field}")

public List<testentity> getWithSort(@PathVariable String field) {

return s.getSorted(field);

}

// pagination

@GetMapping("/product/{offset}/{pageSize}")

public List<testentity> productsWithPagination(@PathVariable int offset, @PathVariable int pageSize) {

return s.getWithPagination(offset, pageSize);

}

}

ENTITY:

package com.example.color.testentity;

import jakarta.persistence.\*;

@Entity

@Table(name="Employee")

public class testentity {

@Id

private int employeeid;

private String firstname;

private String lastname;

private String email;

private long mobile;

private String address;

public int getEmployeeid() {

return employeeid;

}

public void setEmployeeid(int employeeid) {

this.employeeid = employeeid;

}

public String getFirstname() {

return firstname;

}

public void setFirstname(String firstname) {

this.firstname = firstname;

}

public String getLastname() {

return lastname;

}

public void setLastname(String lastname) {

this.lastname = lastname;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public long getMobile() {

return mobile;

}

public void setMobile(long mobile) {

this.mobile = mobile;

}

public String getAddress() {

return address;

}

public void setAddress(String address) {

this.address = address;

}

public testentity(int employeeid, String firstname, String lastname, String email, long mobile, String address) {

super();

this.employeeid = employeeid;

this.firstname = firstname;

this.lastname = lastname;

this.email = email;

this.mobile = mobile;

this.address = address;

}

public testentity(){

}

public String toString() {

return "employeeid="+employeeid+"firstname="+firstname+"lastname="+lastname+"email="+email+"mobile="+mobile+"address="+address;

}

}

REPOSITORY:

package com.example.color.testrepos;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.stereotype.Repository;

import com.example.color.testentity.testentity;

@Repository

public interface testrepos extends JpaRepository<testentity,Integer>{

}

SERVICE:

package com.example.color.testservices;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.data.domain.Page;

import org.springframework.data.domain.PageRequest;

import org.springframework.data.domain.Sort;

import org.springframework.stereotype.Service;

import org.springframework.web.bind.annotation.PathVariable;

import com.example.color.testentity.testentity;

import com.example.color.testrepos.testrepos;

@Service

public class testservices {

@Autowired

testrepos r;

public String add(testentity m) {

r.save(m);

return "Added";

}

public List<testentity> getDetails() {

return r.findAll();

}

public testentity updateDetails(testentity e1) {

return r.saveAndFlush(e1);

}

public void deleteDetails(testentity e2) {

r.delete(e2);

}

public List<testentity> getSorted(String field) {

return r.findAll(Sort.by(Sort.Direction.ASC,field));

}

public List<testentity> getWithPagination(@PathVariable int offset, @PathVariable int pageSize) {

Page<testentity> page =r.findAll(PageRequest.of(offset, pageSize));

return page.getContent();

}

}

OUTPUT:

